A Scheduling Method of Data Processing Task for Burst Mass Data

Tinghui Liu [1, a], Dewen Wang [2, b]

[1]School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China;

[2]School of Control and Computer Engineering, North China Electric Power University, Baoding 071003, China.

[a]Litnguhiwt@163.com, [b] wdewen@gmail.com

Abstract

The development of informatization and intelligentialize is speeding up in the whole world, and the scale of data is also increasing rapidly. Facing the influx of burst mass data, due to the constraints of cloud cluster node processing capacity, the processing node can't complete the data processing in time, and resulting in data accumulation congestion, which makes the node processing capacity to further decline. A scheduling method of data processing task is an important way to solve this problem. In this paper, a scheduling method for burst mass data (MDSS) is proposed. The cloud cluster is divided into three layers, and the central nodes are set up to manage the node resources and dispatch the data processing tasks. The scheduling scheme of multi-level cache is implemented on each node, and the appropriate scheduling nodes are selected by load measurement parameters, which could distributed the processing pressure of nodes .Finally ,the experiment is simulated on the CloudSim platform, which compares the change of the task completion time and cluster load, before and after adopting this method, and analysis the performance of EDF,HVF and DVD algorithm .The results that the method improve task completion obviously, and distribute the cluster load, which would improve the utilization ratio of cluster resource.

Keywords

Burst mass data; Data processing task; Scheduling method; Load balance.

1. Introduction

With the rapid development of information technology, the technology of the global intelligence and information also improve quickly, and leads to larger scale data information[1,2]. Facing the ever-increasing scale of data, cloud computing and big data technology have more extensive applications, and also bring convenience to the analysis and processing for big data[3]. Cluster is connected to a large number of server nodes through the network, which can provide high-performance support services for data processing, analysis and mining. In the face of massive data, the processing ability of a single node is limited, which is difficult to complete the data processing task in-time , and leads to the accumulation of data on the node .The problem will affect the performance of cluster nodes, resulting in more serious blockage or data missing problem.

It's an important way to solve the problem of mass data problem by cloud computing and big data technology[4,5]. The increasingly scale of data information also requires more computing resources to process and analyze data. In order to analyze the data processing and improve the utilization efficiency of the cluster nodes in a limited time, it is necessary to dispatch and distribute the data processing tasks, which can effectively distribute the cluster load and make reasonable use of the resources of different nodes, and also avoid the large number of data processing tasks Stacked in a single node, while other nodes are free of resources.

The scheduling algorithm for data processing tasks is an important measures to solve the problem of data congestion. An improved weighted queue model is proposed in [6] to minimize the latency of low priority tasks and improve the quality of service , in the case of ensuring high priority queue transmission quality. In [7], to distribute the intelligent substation's various processing tasks , which takes the overall effect value of the power quality of service as the goal, and can meet the service requirements in most cases. The two algorithms are static priority scheduling, without considering the impact of time variation on task priority, and can't fully meet the scheduling requirements of burst mass data processing task. Aiming at the information physics system, a middleware of real time data distribution service is proposed in [8], which can solve the problem of transmitting data reliably and timely in high unstable condition.

Based on data stream dispatching, a real-time data exchange middleware is designed and implemented for the efficiency of large scale data exchange across the network for power information systems.

In order to solve the scheduling problem in massive data, this paper proposes a data processing task scheduling method for burst mass data (MDSS), which can distribute data processing task in cluster nodes, according to the load of the central processing node, and also can improve the efficiency of cluster, and reduce data loss and congestion.

2. Node Model Building

2.1 Cluster Hierarchical Model

Aiming at the problem of insufficient node capacity in burst massive data processing, a data scheduling processing node model is proposed in this paper.

Firstly, the cluster node is divided into several sub-clusters, and then each sub-cluster selects the center node. The sub-hub node records the resource status information of the sub cluster , and assists in allocating the data processing tasks. In addition to the cluster hub nodes, there also has a central control node that could coordinate the control of each sub-cluster.

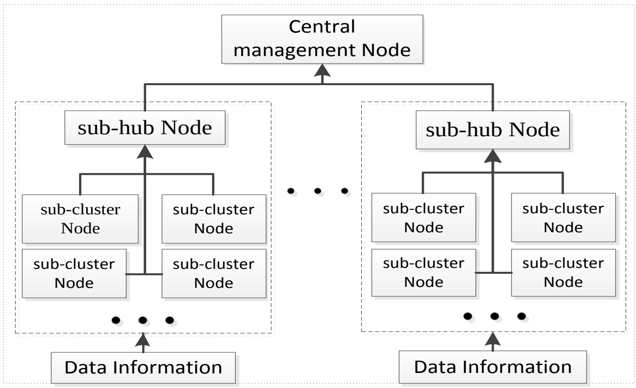


Fig.1 Cluster hierarchical scheduling model

As Fig.1, it shows the hierarchical model of the cluster, the coordination of the sub-clusters is performed by the sub-hub nodes when the data information arrives at each sub-cluster. When the calculated pressure of a node increases, by the node resource records in the sub-hub node, which can select the corresponding scheduling node and distribute it to other nodes in the sub-cluster. When a sub cluster load is too high, the data is transferred remotely through the central management node to other sub clusters, and also can improve the resource utilization of the whole cluster.

2.2 Page Multilevel Schleduling Model in Node

On each sub-cluster node, there are three queues: ordinary task queue, remote interactive queue, and template queue. The processing of data information on nodes is called data processing task, and the status of data processing tasks are divided into waiting state, execution state and migration state.

(1) Wait state: the state of the data information waiting for processing in the ordinary task queue on the sub-cluster node;

(2) Execution state: the task in the waiting state gets the processor resource for analysis and processing;

(3) Migration state: when the number of tasks on the node reaches a certain standard, the part of the data task to be processed is transferred to other nodes.

In the sub cluster nodes, several common task queues are initialized, and the data processing tasks are entered into the ordinary task queue, and processed by the priority of task processing[9]. There are many kinds of priority algorithms available, which can be chosen by the actual application situation, and commonly used algorithm such as HVF, EDF and DVD[10,11,12]. The remote interactive queue mainly performs the task interaction among the nodes. When a node has too many processing tasks and large computing pressure, and move the task to the surrounding free node. The template queue is a template for the other two queues. When the node has remaining resources, but the existing queue is full, the other two queues can be quickly generated.

The migration process of data processing tasks on the queue is shown in Fig.2.

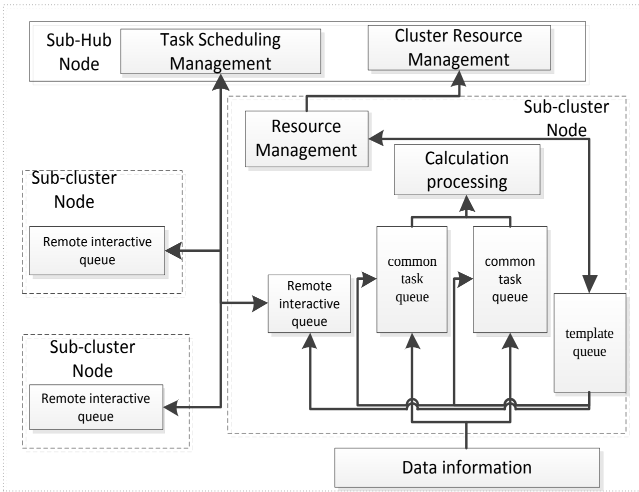


Fig.2 Task processing migration process on node queue

3. Node Selection Resources Parameter Algorithm for Scheduling

The scheduling node's selection will be divided into two parts. The first part is migrate data processing task in the sub cluster, and will be scheduled by central node in sub-cluster. The status of each sub-cluster node will be expressed by different resources, such as CPU utilization, memory usage, and so on. The resource status of the nodes about the whole sub cluster could be represented by the following matrix:

$$R = \begin{Bmatrix} R_1 \\ R_2 \\ \vdots \\ R_n \end{Bmatrix} \qquad (1)$$

Where: R_i is the status of node's resource. R is the resource state express of the sub cluster.

The second part is to choose the destination cluster when the task migration is carried out in different sub clusters. When selecting a destination sub cluster, you need to measure the overall resource status of the different clusters and then select the appropriate migration cluster. After the cluster is selected, select the corresponding node from the cluster.

The total amount of resources in nodes is different, and the ability to process data tasks is also different. Therefore, in the cluster, there has a measurement parameter for each node: Expected number of connection tasks, as follow expression.

$$E_i = \frac{\sum_{1}^{N} e_i}{\sum_{1}^{N} R_i} \cdot R_i \tag{2}$$

Where: E_i is the expected connection task of node, e_i is the actual number of node connections, R_i is resource state of each node.

At the same time, in order to measure the load balancing degree of the cluster, the load balancing difference parameter is introduced as follow,

$$F_h = F_{\max} - F_{avg} \tag{3}$$

Where: F_h is difference between maximum load and average value in the cluster, $F_{\max}$ is the maximum load in the cluster, and F_{avg} is the average load value of the cluster.

4. Experimental Testing and Analysis

The experimental environment is based on the CloudSim simulation platform, which is basic condition of Algorithm design and verification[13]. In the experiment, compared the performance with before and after the MDSS method used. Meanwhile, it also tests the change of algorithm performance before and after adopting different algorithms in MDSS method. In the design of experiment, the first part compares the completion time and the load balance difference , with the same number of tasks. In the second part, comparing with the change of deadline miss rate, analyze the performance of different scheduling algorithms.

The configuration of node in the experiment is shown in table 1.

Table 1. Node configuration

Host Type	0	1	2
MIPS	6000	8000	5000
Cores	1	1	2
Ram	512	512	1024
Storage	100000	100000	100000
Bandwidth	10000	10000	10000

Experiment 1. The performance of cluster before and after adopting MDSS method.

(1) Task completion time comparison

In the experiment, the cluster of 50 nodes is simulated, and the cluster is divided into three sub clusters. At the same time, the data processing task is reached from 1000 to 8000, and the completion time under different states is calculated respectively.

As Fig.3 shows, Task completion time increases with the number of data tasks increasing, while the accumulation of tasks to be processed at the node, and will consume memory resources. Memory consumption will further reduce the processing capacity of nodes, resulting in increased processing time. In the MDSS method, when the arrival data processing task is more in the node, it will disperse the task to the surrounding nodes, and deal with the task, making the trend of the completion time of the task more slowly.

(2)Load balancing difference comparison.

In the load balancing experiment, the data processing task changes from 1000 to 8000, and the difference between cluster load and average load is used to measure the load balance of cluster.

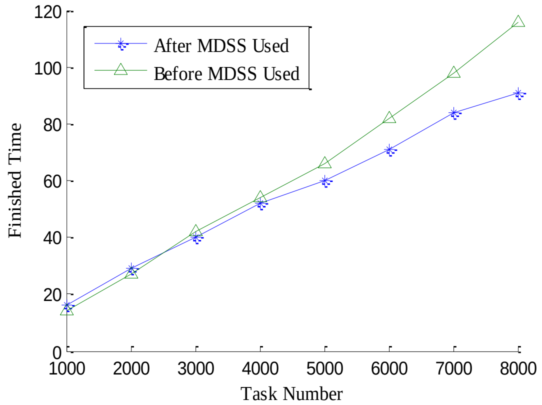


Fig.3 Task completion time comparison

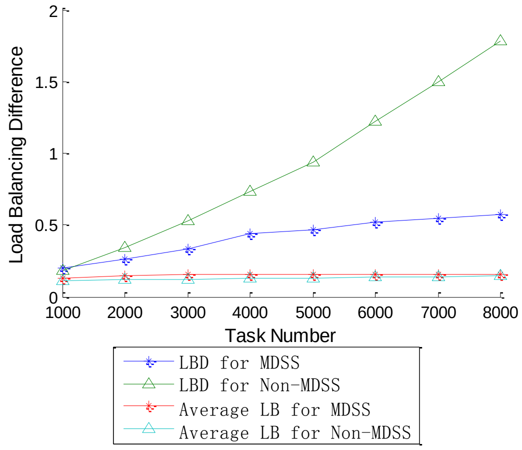
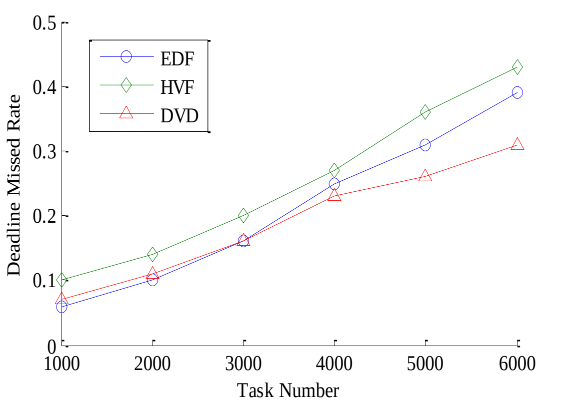


Fig.4 Load balancing difference comparison

As Fig.4 shows, the trend of maximum load and average value of cluster is respectively analyzed, before and after adopting MDSS method. Before MDSS is used, when a data processing task arrives, it is handled by a single node. When the number of tasks arrives is larger, the processing pressure of nodes increases, which will affect the processing performance of nodes. In the MDSS method, by measuring the status of node load, scheduling data processing tasks to other nodes, which make the load of the cluster more balanced, and improve the utilization rate of cluster resources, increase processing capacity.

Experiment 2. Comparison of performance improvement of different algorithms.



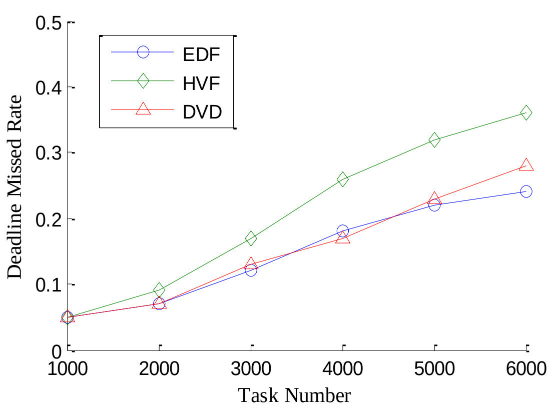


Fig.5 (1) Comparison of different algorithms before MDSS method (left); (2) Comparison of different algorithms after MDSS method (right)

As Fig.5(1) and(2)show, before and after using the MDSS method, the deadline miss rate of three algorithms under different task number is compared. In Fig.5(1), the number of tasks handled at the initial time is small, and the error rate of each algorithm is not much difference. With the increase of the number of tasks, the HVF algorithm has the greatest priority for the maximum value task, so the sensitivity of the time requirement is low, and the deadline miss rate changes greatly. When dealing with fewer tasks, EDF algorithm can achieve better scheduling effect, but when the task is more, the system load increases, the scheduling performance will decline rapidly. DVD algorithm needs to dynamically calculate the priority of tasks, when the task is less, because the calculation consumes part of the resources, so it is a little higher than the EDF algorithm miss rate. When the data processing task is increased, the DVD algorithm changes smoothly, and the deadline loss rate increases slowly. In Fig.5(2), after using the MDSS method, when the data processing task is more, there will migrate tasks to surrounding nodes, dispersed computing pressure, and then the deadline miss rate has decreased significantly.

5. Conclusion

Due to the capacity node processing constraints, for burst mass data processing problems, if not processing in time, it may lead to data loss of the problem. The scheduling of data processing tasks can improve the completion rate of tasks and effectively reduce the processing delay. In this paper, a method for dealing with burst mass data is proposed, the hierarchical management of the cluster is used to coordinate the management of the node resources. Multi queue buffer for scheduling management is performed on nodes, and the appropriate scheduling nodes are selected to distribute tasks when the computing pressure of single node too much. In the simulation experiment on CloudSim platform, after use the MDSS method, it's obviously reduce the task completion time and the deadline miss rate. In the subsequent research, the scheduling algorithm that takes into account both the deadline and the task value is considered, and further optimization of the method is needed.

References

[1] Cheng Xueqi, Jin Xiaolong,et al. Survey on Big Data System and Analytic Technology[J]. Journal of Software, 2014(9):1889-1908.

[2] Sun Dawei, Zhang Guangyan, et al. Big Data Stream Computing： Technologies and Instances [J].Journal of Software,2014, 25(4):839-862.

[3] Ma F, Luo X, Litvinov E. Cloud Computing for Power System Simulations at ISO New England—Experiences and Challenges[J]. IEEE Transactions on Smart Grid, 2016, 7(6): 2596-2603.

[4] Zhong Z, Chen K, Zhai X, et al. Virtual machine-based task scheduling algorithm in a cloud computing environment[J]. Tsinghua Science and Technology, 2016, 21(6): 660-667.

[5] GONG Shilin, WANG Chen, FENG Yanzhao et al. A Smart Substation LAN Communication Queue Scheduling Strategy Based on Improved WFQ Algorithm[J]. Automation of Electric Power Systems, 2015,(04):76-81.

[6] WANG Dewen, LIU Yang. A Task Scheduling Strategy of Cloud Data Center in Electric Power Corporation[J]. Automation of Electric Power Systems, 2014,(08):61-66+97.

[7] Kang W, Kapitanova K, Sang H S. RDDS: A Real-Time Data Distribution Service for Cyber-Physical Systems[J]. IEEE Transactions on Industrial Informatics, 2012, 8(2):393-405.

[8] Cao Jian, Liu Qiong. Firm real-time data-transmitting system based on data stream-transmitting mechanism[J]. journal of Computer Applications. 2016, 36(3):596-600.

[9] Lee J. Time-Reversibility for Real-Time Scheduling on Multiprocessor Systems[J]. IEEE Transactions on Parallel and Distributed Systems, 2017, 28(1): 230-243.

[10] Aldarmi S A, Burns A. Dynamic value-density for scheduling real-time systems[C]//Real-Time Systems, 1999. Proceedings of the 11th Euromicro Conference on. IEEE, 1999: 270-277.

[11] XIA Jiali, CHEN Hui, YANG Bing. A Real-Time Tasks Scheduling Algorithm Based on Dynamic Priority[J]. Chinese Journal of Computers. 2012,(12):2685-2695.

[12] Calheiros R N, Ranjan R, De Rose C A F, et al. Cloudsim: A novel framework for modeling and simulation of cloud computing infrastructures and services[J]. arXiv preprint arXiv:0903.2525, 2009.